

USER AND TECHNICAL GUIDE

UCJIS XMLConduit Web Service

Revised 10/13/2005

Table of Contents

1.0 BACKGROUND	1
2.0 WEB SERVICE IMPLEMENTATION: UCJIS XMLCONDUIT.....	1
2.1 Overview.....	1
2.2 Technical Details	1
2.2.1 XML.....	1
2.2.2 Web Service	1
2.2.2.1 Creating a Proxy	1
2.2.2.2 Methods	3
2.3 LINKS	3
2.3.1 Core Technologies.....	3
2.3.1.1 WSDL Specification	4
2.3.1.2 Java Web Services Developer Pack	4
2.3.1.3 SOAP	4
2.3.2 General Information	4

1.0 Background

Web Services are designed to overcome the problems associated with proprietary network protocols. A core set of specifications that together make up Web Services are XML, SOAP, and WSDL. The Extensible Markup Language (XML) is a W3C standard for defining data using a simplified syntax similar to the Standard Generalized Markup Language (SGML). XML is fast becoming the de facto standard for defining data and documents used in Internet-based transactions, and it is used to define SOAP.

Simple Object Access Protocol (SOAP) is a specification that incorporates the most important features of the aforementioned protocols, e.g., Distributed Component Object Model (DCOM) and Remote Method Invocation (RMI), into a light-weight XML message that can be used to transmit data over HTTP. SOAP, however, is not limited to just HTTP; it can be used in combination with a variety of protocols, depending upon the business requirements.

Web Services Description Language (WSDL) is an XML format for describing Web Services, as a set of endpoints – or ports – which operate on messages sent over the network. This document assumes that the ongoing reader already has a sufficient background in the web service technologies and implementation options available in his or her development tool(s) of choice.

2.0 Web Service Implementation: UCJIS XMLConduit.

This section describes how UCJIS is using Web Services at the present time.

2.1 Overview

The Department of Public Safety, Management Information Services (DPSMIS) is currently developing a web service to improve information sharing among the numerous terminals connected to the UCJIS network. The technologies employed in the development of this program are:

- XML 1.0
- SOAP 1.1
- WSDL 1.2

2.2 Technical Details

This section includes all of the elements required to design, build, and integrate with the UCJIS web service: XMLConduit.

2.2.1 XML

The body of an XMLConduit request is an XML document. For more information on the structure of these XML documents, please see the transaction specific documentation available from DPSMIS, i.e., Mike Sadler: msadler@utah.gov.

2.2.2 Web Service

2.2.2.1 Creating a Proxy

The easiest way to create a client for the XMLConduit web service is to extract the Web Services Description Language (WSDL) file from the server and generate source code stubs or a proxy. A WSDL document is an XML file for describing a Web

Service's interface and its set of endpoints – or ports – which operate on messages sent over the network. The WSDL can be fed into an implementer's tool(s) of choice to create the stub code needed to statically bind on the server's interface within a distributed environment.

An abbreviated and preliminary version of the XMLConduit WSDL is illustrated in the following table. Note however, that it is best to extract the current WSDL directly from the server.

UCJIS XMLConduit Web Service WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="XMLConduit" targetNamespace="urn:XMLConduit/wsdl" xmlns:tns="urn:XMLConduit/wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  <types/>
  <message name="XMLConduitRPC_processXMLDocument">
    <part name="String_1" type="xsd:string"/>
  </message>
  <message name="XMLConduitRPC_processXMLDocumentResponse">
    <part name="result" type="xsd:string"/>
  </message>
  <message name="XMLConduitRPC_processEncodedDocument">
    <part name="String_1" type="xsd:string"/>
  </message>
  <message name="XMLConduitRPC_processEncodedDocumentResponse">
    <part name="result" type="xsd:string"/>
  </message>
  <portType name="XMLConduitRPC">
    <operation name="processXMLDocument" parameterOrder="String_1">
      <input message="tns:XMLConduitRPC_processXMLDocument"/>
      <output message="tns:XMLConduitRPC_processXMLDocumentResponse"/>
    </operation>
    <operation name="processEncodedDocument" parameterOrder="String_1">
      <input message="tns:XMLConduitRPC_processEncodedDocument"/>
      <output message="tns:XMLConduitRPC_processEncodedDocumentResponse"/>
    </operation>
  </portType>
  <binding name="XMLConduitRPCBinding" type="tns:XMLConduitRPC">
    <operation name="processXMLDocument">
      <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
          namespace="urn:XMLConduit/wsdl"/>
      </input>
      <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
          namespace="urn:XMLConduit/wsdl"/>
      </output>
      <soap:operation soapAction="" />
    </operation>
    <operation name="processEncodedDocument">
      <input>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
          namespace="urn:XMLConduit/wsdl"/>
      </input>
      <output>
        <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
          namespace="urn:XMLConduit/wsdl"/>
      </output>
      <soap:operation soapAction="" />
    </operation>
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/>
  </binding>
  <service name="XMLConduit">
    <port name="XMLConduitRPCPort" binding="tns:XMLConduitRPCBinding">
      <soap:address location="http://webservices.ucjis.utah.gov/XMLConduit/XMLConduit"/>
    </port>
  </service>
</definitions>
```

The XMLConduit Web Service WSDL, as of 01/26/2004, is located at the following address: <http://webservices.ucjis.utah.gov/XMLConduit/XMLConduit?WSDL> or a request for the current WSDL can be made from msadler@utah.gov.

2.2.2.2 Methods

The XMLConduit web service currently supports two method as enumerated below.

Method Name:	processXMLDocument (Deprecated)
Method Argument(s):	(XML string)
Return Value	(XML string)
Description	This operation has been deprecated and should not be implemented.
Method Name:	processEncodedDocument
Method Argument(s):	(Base64 encoded XML string)
Return Value	(Base64 encoded XML string)
Description	<p>The strings constituting the input parameter and return value of processEncodedDocument() are Base64 encoded XML documents whose semantics are beyond the scope of this document and are specific to the transaction in context. Collaborating systems that use the XMLConduit web service will determine the content of input and return data based on published standards, best practices, and the Justice XML Specification. Most often, however, logical branching can be performed on the DocumentDescriptor element and its associated attributes, type and class, contained in most UCJIS transactions.</p> <p>Note: According to the Justice XML Specification, binary attachments are transmitted within XML documents, base64 encoded. They are not transmitted as SOAP, DIME or other multi-part form attachments</p>

2.3 LINKS

Additional information is available by connecting to the web sites listed in the sections that follow.

2.3.1 Core Technologies

The links below provide additional information as specified by the section headings.

2.3.1.1 WSDL Specification

<http://www.w3.org/TR/wsdl>

2.3.1.2 Java Web Services Developer Pack

<http://java.sun.com/webservices/downloads/webservicespack.html>

2.3.1.3 SOAP

Microsoft SOAP Toolkit (VB6) Implementation:

<http://msdn.microsoft.com/library/default.asp?url=/downloads/list/websrv.asp>

Microsoft .NET Implementation:

<http://msdn.microsoft.com/downloads/default.asp?url=/downloads/sample.asp?url=/msdn-files/027/002/108/msdncompositedoc.xml>

J2EE Implementation:

<http://xml.apache.org/axis/index.html>

2.3.2 General Information

Understanding GXA (Microsoft's Global XML Architecture):

<http://msdn.microsoft.com/webservices/understanding/gxa/default.aspx?pull=/library/en-us/dngxa/html/understandgxa.asp>